

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

ORACLE AMERICA, INC.

Plaintiff,

v.

GOOGLE INC.

Defendant.

Case No. CV 10-03561 WHA

SUMMARY AND REPORT OF EREZ LANDAU

**SUBMITTED ON BEHALF OF PLAINTIFF
ORACLE AMERICA, INC.**

IV. PERFORMANCE BENCHMARK AND TESTING ANALYSIS OF U.S. PATENT NO. 7,426,720

A. Overview

25. The baseline for these experiments was the Froyo release of Android, pulled from the Google git repository with the Google repo commands

```
$ repo init -u git://android.git.kernel.org/platform/manifest.git -b froyo
```

```
$ repo sync
```

to initialize and sync from the Google repository. I made two copies of the repository that contained the Android system: (1) an unmodified copy to serve as a baseline for comparison and (2) a version with the zygote functionality that Oracle accuses of infringing the '720 patent disabled.

26. I disabled the use of the accused functionality in Android as discussed in Appendix A.

27. The modification for experiments were restricted to two files:

- a. frameworks/base/core/java/com/android/internal/os/RuntimeInit.java and
- b. frameworks/base/core/jni/AndroidRuntime.cpp.

In summary, the zygote process which is the initial (master) virtual machine which manages the virtual machine cloning (and thus sharing) was modified to call the new 'exec' method for the new created process (clone), so it re-initializes the process as a fresh new virtual machine (over running whatever was inherited from the master virtual machine) thus disabling the '720 patent. The modifications are detailed and shown in Appendix A.

28. All workspaces were compiled and loaded to the device following the directions described at Google's <http://source.android.com/source/initializing.html>.

29. The benchmarks used to test the performance of the workspaces were all Google standard applications available in the repository, as listed above.

30. I tested the performance of the repositories by running them on the HTC Nexus One with the two workspaces installed on two separate phones: (1) one phone to test the

unmodified Android source code (where the functionality that Oracle accuses of infringing the '720 patent is practiced and not disabled – labeled “Original (copy-on-write)” in Exhibit A) and (2) another phone to test the modified Android source code (disabling the functionality that Oracle accuses of infringing the '720 patent – labeled “No VM Sharing” in Exhibit A).

31. I ran all benchmark applications on both phones.

32. The accompanying charts attached as Exhibit A records the results of runs of a standard phone startup which launch all the default applications.

B. Mtask ClassLoading Test

33. I combined HelloWorld and MtaskClassLoading tests in a single application with Google HelloActivity which says “hello” and does the class loading. That is to say, the MtaskClassLoading test says “hello” and does class loading of a given class list. The test is actually a modified version of android HelloActivity sample available under folder ‘development/samples/HelloActivity’ adding a call to MtaskClassLoading.main() method to do the class loading.

34. The class list is made of Android relevant preloaded classes. That is, the list of classes to load was taken from the Android preloaded class list.

35. Test code is attached as Exhibit B.

36. Logcat dumps are attached as Exhibits C and D.

V. CONCLUSION

37. The performance benchmark testing shows as much as 40% memory savings (Android uses 51MB instead of 86MB) and also saves .10 second per application launch time. In other words, disabling the functionality that Oracle accuses of infringing the '720 patent increases the memory consumption by 70% (because 86 is 1.7 times 51).

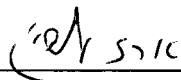
38. Memory savings were 2MB per each running application. This becomes more significant when executing many small widgets with a small amount of required unique memory consumption. Moreover, the results demonstrate that for applications that need more of the

preloaded classes, the launch time saving will be substantially more than 0.1 seconds and could increase up to 1.5 seconds.

39. The ability to reduce the per-application memory payload is significant because it allows a true multiple applications environment.

I declare under penalty of perjury that the foregoing is true and correct.

Dated: August 8, 2011



Erez Landau